



TITLE:

最小消去多項式を用いた行列スペクトル分解計算の並列化 (Computer Algebra : Design of Algorithms, Implementations and Applications)

AUTHOR(S):

小原, 功任; 田島, 慎一

CITATION:

小原, 功任 ...[et al]. 最小消去多項式を用いた行列スペクトル分解計算の並列化 (Computer Algebra : Design of Algorithms, Implementations and Applications). 数理解析研究所講究録 2012, 1815: 21-28

ISSUE DATE:

2012-10

URL:

<http://hdl.handle.net/2433/194575>

RIGHT:

最小消去多項式を用いた 行列スペクトル分解計算の並列化

小原功任

KATSUYOSHI OHARA

金沢大学理工研究域数物科学系*

田島慎一

SHINICHI TAJIMA

筑波大学大学院数理物質科学研究科†

1 行列スペクトル分解

正方行列 A の固有値を $\lambda_1, \dots, \lambda_m$ とし, 固有値 λ_i に関する不変部分空間を V_i とする。よく知られているように、このとき互いに可換な行列 P_1, \dots, P_m, D が存在して

$$E = P_1 + \dots + P_m,$$

$$A = \lambda_1 P_1 + \dots + \lambda_m P_m + D,$$

P_i は V_i への射影行列,

D は巾零行列,

となる。この $\{P_1, \dots, P_m; D\}$ を行列 A のスペクトル分解という。

本研究の目的は、 \mathbf{Q} 上の行列のスペクトル分解をできるだけ高速に exact に計算することである。そのために、数学的にはレゾルベントに関する留数解析を、計算技術としては並列計算の手法を用いる。また、 \mathbf{Q} 上の行列のスペクトル分解であるから、一般に行列 P_i, D は固有値 λ_i の多項式または有理式を成分とする。このときに冗長な表現で求めてしまうと、一見して高速に計算できたように思えても、スペクトル分解の後処理としての simplification あるいは一意化の手間が馬鹿にならずかえって時間がかかってしまう。したがって本研究では最小次数の多項式を成分とするような結果を求めるものである。本稿でははじめにスペクトル分解アルゴリズムの概略を、その後に今回工夫を行った最小消去多項式を利用した高速計算法の概略を述べ、最後に並列計算アルゴリズムについて説明する。また、われわれのアルゴリズムは計算代数システム Risa/Asir に実装しているが、そのタイミングデータについても述べる。

2 レゾルベントと射影行列

定義 1. 正方行列 A に対して、行列値複素関数 $R(z) = (zE - A)^{-1}$ を A のレゾルベントという。

*ohara@air.s.kanazawa-u.ac.jp

†tajima@math.tsukuba.ac.jp

簡単に分かるように、レゾルベント $R(z)$ は \mathbb{C} 上の行列値有理関数となる。また $R(z)$ の極は A の特性多項式 $\chi(z) = \det(zE - A)$ の根、すなわち固有値である。レゾルベントについては次の定理がよく知られている。

定理 1. A の固有値を $\lambda_1, \dots, \lambda_m$ とする。 λ_i におけるローラン展開

$$R(z) = \cdots + \frac{1}{(z - \lambda_i)^2} D_i + \frac{1}{z - \lambda_i} P_i + \cdots$$

は、 A のスペクトル分解 $\{P_1, \dots, P_m; \sum_{i=1}^m D_i\}$ を与える。

したがってスペクトル分解を求めるには、 $R(z)$ の固有値のまわりでのローラン展開について調べればよいことになる。特に、コーシーの積分定理から次がしたがう。

定理 2. 固有値 $z = \lambda_i$ のまわりを反時計回りに一周まわる閉曲線を C_i とする。このとき、 λ_i に対する射影行列 P_i は、周回積分 $P_i = \frac{1}{2\pi\sqrt{-1}} \int_{C_i} R(z) dz$ によって与えられる。

また、レゾルベントの有理関数としての構造については次の定理が得られる。

定理 3. 多項式 $f(z) \in \mathbb{C}[z] \setminus \{0\}$ が $f(A) = O$ を満たせば、レゾルベントは

$$R(z) = \frac{1}{f(z)} \Psi_f(zE, A), \quad \text{ただし} \quad \Psi_f(x, y) = \frac{f(x) - f(y)}{x - y} \in \mathbb{C}[x, y].$$

と表される。

例 1. 行列 $A = \begin{pmatrix} 0 & 4 & 0 \\ -1 & 4 & 0 \\ 0 & 0 & 3 \end{pmatrix}$ の最小多項式は $\pi(x) = (x - 2)^2(x - 3)$ であり、固有値は $\lambda_1 = 2, \lambda_2 = 3$ である。このとき、 A のレゾルベントは

$$R(z) = \frac{1}{(z - 2)^2} (-6E + 5A - A^2) + \frac{1}{z - 2} (-3E + 4A - A^2) + \frac{1}{z - 3} (4E - 4A + A^2)$$

と書かれ、その係数行列

$$\begin{aligned} P_1 &= -3E + 4A - A^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & P_2 &= 4E - 4A + A^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ D_1 &= -6E + 5A - A^2 = \begin{pmatrix} -2 & 4 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & D_2 &= O \end{aligned}$$

は A のスペクトル分解 $\{P_1, P_2; D_1 + D_2\}$ を与える。

例 2. 行列

$$A = \begin{pmatrix} 0 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

の最小多項式は $\pi(x) = (x^2 - 2)^2$ であり、固有値は $\lambda_1 = \sqrt{2}$, $\lambda_2 = -\sqrt{2}$ である。このとき、 A のスペクトル分解 $\{P_1, P_2; D_1 + D_2\}$ は

$$P_i = \begin{pmatrix} \frac{1}{2} & \frac{\lambda_i}{2} & 0 & \frac{\lambda_i}{8} \\ \frac{\lambda_i}{4} & \frac{1}{2} & -\frac{\lambda_i}{16} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{\lambda_i}{2} \\ 0 & 0 & \frac{\lambda_i}{4} & \frac{1}{2} \end{pmatrix}, \quad D_i = \begin{pmatrix} 0 & 0 & \frac{\lambda_i}{8} & \frac{1}{4} \\ 0 & 0 & \frac{1}{8} & \frac{\lambda_i}{8} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (i = 1, 2)$$

で与えられる。

3 レゾルベントの留数解析とネーター作用素

定理 3 より行列 A を消去する多項式があれば、レゾルベントを具体的に書き下すことができるが、計算量的観点からは、次数が小さな多項式を選ぶべきである。したがってレゾルベントは最小多項式 $\pi(z)$ を用いて $R(z) = \frac{1}{\pi(z)} \Psi_\pi(zE, A)$ と表されることになる。さらに、定理 2 より、レゾルベントの周回積分で射影行列は表されるので、最小多項式で表されたレゾルベントについて、周回積分を exact に求めることができ、射影行列の計算はできることになる。そのために代数的局所コホモロジー群とネーター作用素の概念を導入する。ネーター作用素計算の Risa/Asir への実装については、[4] を参照のこと。本節では、射影行列 P_i についてのみ説明するが、巾零部分 D_i も同様に計算可能である。

$f(z) \in \mathbf{Q}[z]$ を既約多項式とする。集合 $V(f) = \{z \mid f(z) = 0\}$ に特異点をもつ有理関数の全体 $\mathbf{Q}[z, f^{-1}]$ に同値関係 \equiv を次のように入れる。

$$r_1(z) \equiv r_2(z) \iff r_1(z) - r_2(z) \in \mathbf{Q}[z]$$

この同値類を $[r_1]$ で表し、 $V(f)$ に台をもつ代数的局所コホモロジー類という。また、この同値類の全体を $V(f)$ に台をもつ代数的局所コホモロジー群という。このとき、与えられた有理関数 $g(z)/f(z)^\ell \in \mathbf{Q}[z, f^{-1}]$ に対して、次の多項式係数 $\ell - 1$ 階微分作用素 T を計算することができる ([3], [4])。

$$\left[\frac{g(z)}{f(z)^\ell} \right] = T \left[\frac{f'(z)}{f(z)} \right], \quad T = \sum_{k=0}^{\ell-1} (-\partial_z)^{\ell-k} t_k(z) \in \mathbf{Q}\langle z, \partial_z \rangle.$$

この T をネーター作用素という。より一般には、 $V(f_1 \cdots f_m)$ に台をもつ代数的局所コホモロジー群を考えることができ、有理関数 $r(z) \in \mathbf{Q}[z, f_1^{-1}, \dots, f_m^{-1}]$ の代数的局所コホモロジー類に対して、既約成分ごとのネーター作用素表示

$$[r(z)] = T_1 \left[\frac{f'_1(z)}{f_1(z)} \right] + \cdots + T_m \left[\frac{f'_m(z)}{f_m(z)} \right], \quad T_i \in \mathbf{Q}\langle z, \partial_z \rangle$$

が求まる。

行列 A の固有値 λ_i に対する射影行列 P_i を求めるために、ネーター作用素が使えることを示そう。いま、 A の最小多項式を $\pi(z) = f_1(z)^{\ell_1} \cdots f_m(z)^{\ell_m}$ とする。 $f_i(z)$ は固有値 λ_i の最小多項式である。まず、ネーター作用素

$$\left[\frac{1}{\pi(z)} \right] = T_1 \left[\frac{f'_1(z)}{f_1(z)} \right] + \cdots + T_m \left[\frac{f'_m(z)}{f_m(z)} \right]$$

を求める。ネーター作用素 $T_i = \sum_{k=0}^{\ell_i-1} (-\partial_z)^{\ell_i-k} t_k(z)$ に対して、随伴作用素 $T_i^* = \sum_{k=0}^{\ell_i-1} t_k(z) (\partial_z)^{\ell_i-k}$ は容易に得られる。部分積分と留数定理により

$$\begin{aligned} P_i &= \frac{1}{2\pi\sqrt{-1}} \int_{C_i} R(z) dz = \frac{1}{2\pi\sqrt{-1}} \int_{C_i} \Psi_\pi(zE, A) \frac{1}{\pi(z)} dz \\ &= \frac{1}{2\pi\sqrt{-1}} \int_{C_i} \Psi_\pi(zE, A) \sum_{j=1}^m T_j \frac{f'_j(z)}{f_j(z)} dz = \frac{1}{2\pi\sqrt{-1}} \int_{C_i} \Psi_\pi(zE, A) T_i \frac{f'_i(z)}{f_i(z)} dz \\ &= \frac{1}{2\pi\sqrt{-1}} \int_{C_i} \{T_i^* \Psi_\pi(zE, A)\} \frac{f'_i(z)}{f_i(z)} dz = \{T_i^* \Psi_\pi(zE, A)\}|_{z=\lambda_i} \end{aligned}$$

このことから2変数多項式 $\Psi_\pi(z, y)$ に対して、微分作用素 T_i^* を作用させ、得られた多項式 $F_i(z, y) = (T_i^* \Psi_\pi)(z, y)$ についての行列多項式 $F_i(\lambda_i E, A)$ が射影行列 P_i になることが分かった。

4 最小消去多項式を用いた高速計算法とその並列化

4.1 最小消去多項式とスペクトル分解アルゴリズムの関係

定義 2. 正方行列 A とベクトル $v \neq 0$ に対し、 $f(A)v = 0$ を満たす次数最小のモニック多項式を、 v の**最小消去多項式**という。

定理 3 の多項式 $f(z)$ を最小消去多項式で置き換えることで、次の定理が得られる。

定理 4. 第 j 基本ベクトル e_j の最小消去多項式を $\pi_j(z)$ とする。このとき、 A のレゾルベント $R(z)$ について

$$R(z)e_j = \frac{1}{\pi_j(z)} \Psi_j(zE, A)e_j, \quad \text{ただし} \quad \Psi_j(x, y) = \frac{\pi_j(x) - \pi_j(y)}{x - y} \in \mathbb{Q}[x, y].$$

が成り立つ。

A の最小多項式 $\pi(z)$ と最小消去多項式 $\pi_j(z)$ を比較すると、 $\deg \pi_j \leq \deg \pi$ であることは容易に分かる。つまり最小多項式を最小消去多項式に置き換えることで、アルゴリズムを高速化することができる。 P_i を求めるための新しいスペクトル分解アルゴリズムは次のようになる。

アルゴリズム 1.

入力: \mathbb{Q} 上の正方行列 A

出力: スペクトル分解 $\{P_i; D_i\}$

1. A の特性多項式 $\chi(z)$ を求める。
2. 各 e_j の最小消去多項式 $\pi_j(z) = f_1^{\ell_{1j}} \cdots f_m^{\ell_{mj}}$ とその既約分解を求める。
3. $[1/\pi_j(z)]$ のネーター作用素 T_{1j}, \dots, T_{mj} を求める。
4. 多項式 $F_{ij}(z, y) = T_{ij}^* \Psi_j(z, y) \bmod f_i(z)$ を求める。
5. 行列多項式 $P_i e_j = F_{ij}(\lambda_i E, A) e_j$ を求める。
6. P_i を得る。

前節と同様にアルゴリズム 1 における計算量を見積もると、 P_i を求めるためには、 $O(n^2 \deg f_i \sum_{j=1}^n \deg \pi_j)$ となる。 $\frac{1}{n} \sum_{j=1}^n \deg \pi_j \leq \deg \pi$ であるから、アルゴリズム 1 のほうが計算量が小さい。この差が大きければ大きいほど高速化されたことになる (ブロック対角化されている場合など)。

アルゴリズムの並列化の観点から見ると、ステップ 3 については、列番号 j でそれぞれ独立に計算でき、ステップ 4,5 については、既約因子番号 i および列番号 j でそれぞれ独立に計算できることが分かる。さらに後述するようにステップ 2 における最小消去多項式の構成においても並列化技法が使用できる。したがってアルゴリズム 1 は並列化の面からは非常に扱いやすい。

4.2 最小消去多項式導出の逐次的計算法

本小節では、第 j 基本ベクトル e_j に関する最小消去多項式の逐次的計算法について概説する。アルゴリズムの詳細については [8] を参照すること。

特性多項式の既約分解が $\chi(z) = f_1(z)^{\ell_1} \cdots f_m(z)^{\ell_m}$ であれば、最小消去多項式は

$$\pi_j(z) = f_1(z)^{\ell_{1j}} \cdots f_m(z)^{\ell_{mj}}, \quad (0 \leq \ell_{ij} \leq \ell_i)$$

の形になる。つまり、既約因子は分かるため指数ベクトル $(\ell_{1j}, \dots, \ell_{mj})$ が問題である。指数ベクトルは次の手順で求めることができる。

アルゴリズム 2. $k = 1, \dots, m$ および $j = 1, \dots, n$ について次を行う。

1. $\chi_k(z) = \chi(z)/f_k(z)^{\ell_k}$ とおき、 $b_{kj} = \chi_k(A)e_j$ をすべて求める。
2. $b_{kj} = 0$ ならば、 $\ell_{kj} = 0$ である。 $b_{kj} \neq 0$ ならば、順に $f_k(A)$ を左からかけて、 $f_k(A)^p b_{kj} = 0$ となる最小の $p > 0$ が $\ell_{kj} = p$ である。

さて、この方法で最小消去多項式が計算できることは分かったが、問題点は計算量が大きいことである。したがって、確率的アルゴリズムを導入して高速化を図る。確率的アルゴリズムのポイントは、どんなベクトル v に対しても

$$\pi_j(A)e_j = 0 \quad \text{ならば} \quad {}^t v \pi_j(A)e_j = 0$$

が成り立つという事実である。もちろん逆は成り立たないが、乱数ベクトル v をとり、 ${}^t v \pi_j(A)e_j = 0$ で最小消去多項式を探索すれば、ほぼ確率 1 で最小消去多項式を発見できることが期待される。また、 $f_k(A)e_j$ は n 次元ベクトルだが、 ${}^t v f_k(A)e_j$ はスカラー量なので、計算量も $1/n$ となる。

アルゴリズム 3. 乱数ベクトル v を選び、 $k = 1, \dots, m$ について次を行う。

1. $\chi_k(z) = \chi(z)/f_k(z)^{\ell_k}$ とおき、 ${}^t v_k = {}^t v \chi_k(A)$ をすべて求める。
2. $j = 1, \dots, n$ について次を行う。
 ${}^t v_k e_j = 0$ ならば、 $\ell_{kj} = 0$ と置く。0 でなければ、 ${}^t v$ に順に $f_k(A)$ を右からかけて、 ${}^t v_k f_k(A)^p$ を求め、 ${}^t v_k f_k(A)^p b_{kj} = 0$ となる最小の $p > 0$ で $\ell_{kj} = p$ と置く。

3. $j = 1, \dots, n$ について次を行う。

推定した指数ベクトル (「最小消去多項式候補」と呼ぶ) が本当に最小消去多項式になっているか、実際に $\pi_j(A)e_j$ を計算して確認する。最小消去多項式でなければアルゴリズム 2 でやり直し。

ステップ 1 では、すべての v_1, \dots, v_m を求めるには、ベキ乗算 ${}^t u \mapsto {}^t u f_k(A)^{\ell_k}$ が m^2 回必要であるように思えるが、2 分探索の方法を用いれば、およそ $m \log_2 m$ 回で実行可能である。これを詳しく説明しよう。いま、 $S = \{1, \dots, m\}$ の部分集合 I に対して、ベクトル v_I を

$${}^t v_I = {}^t v \prod_{i \in S \setminus I} f_i(A)^{\ell_i}$$

で定める。すると、 $v_k = v_{\{k\}}$ かつ $v = v_S$ である。 S を根とし、singleton $\{k\}$, ($k = 1, \dots, m$) を葉とする二分木を次の規則で定める。

I が親で J, K がその子であるならば、 $I = J \sqcup K$ (disjoint union) である。

例 3. $m = 4$ とする。このとき、集合

$$T = \{\{1, 2, 3, 4\}, \{1, 2\}, \{3, 4\}, \{1\}, \{2\}, \{3\}, \{4\}\}$$

は $S = \{1, 2, 3, 4\}$ を根とし、 $\{k\}$ を葉とする二分木の構造が入る。

さて、 I と J が親子関係にあるとき、ベクトルの親子 v_I, v_J の間にも関係式

$${}^t v_J = {}^t v_I \prod_{i \in I \setminus J} f_i(A)^{\ell_i},$$

がある。よって親ノードの計算結果を用いて子ノードのベクトルが計算できる。二分木を辿ることによって、すべての葉におけるベクトル v_k が求まる。

さらに、 v_k は一度計算しておけば、どの基本ベクトル e_j の計算にも利用できることに注意する。また行列 $\chi_k(A)$ を計算するより、ベクトル ${}^t v_{\chi_k(A)}$ の計算の方がはるかに計算量が小さい。よって、 $\chi_k(A)$ を計算しておいて、再利用を試みるのは計算量的に危険だが、 ${}^t v_{\chi_k(A)}$ ではそれほどでもない。

4.3 最小消去多項式導出の並列計算法

ここでは、アルゴリズム 3 の並列化について議論したい。ステップ 2, 3 は単純なループであり、ステップ 2 は各 j, k について、ステップ 3 は各 j について計算が独立であるから、並列化できることは明らかである。よって問題になるのはステップ 1 である。前節で説明したように、ステップ 1 では二分木を辿りながら v_1, \dots, v_m を求めることができる。このとき親子関係にないノードでは、独立に計算を実行できるが、親子関係にあるときには、計算順序の制御が必要である。数式処理システム Risa/Asir における並列計算の実装には、oh.p パッケージが利用できるので、これにもとづいて二分木を辿る並列計算を説明しよう ([9])。

いま二分木を根の方から辿るような計算をすることとし、子ノードの計算には親ノードの計算結果が必要であると仮定する。親ノードと子ノードが同時に計算されると、不都合が生じるので、同時に計算を行うのは、親子関係にないノードでなければならない。また最初は根の計算の時には、 n 個のプロセスがあっても 1 個だけ計算を行い、他のプロセスは待機することになる。

いま、二分木の各ノードの全体をジョブ集合 T とし、既に計算が終わったジョブの全体、計算中のジョブの全体、残りのジョブの全体をそれぞれ、 T_f, T_b, T_u とする。つまり、 $T = T_f \sqcup T_b \sqcup T_u$ である。また $\emptyset \notin T$ とする。次のような関数 φ を考える。

$$\varphi(T_u, T_f) = \begin{cases} x & (x \in T_u \text{ かつ } \text{parent}(x) \in T_f) \\ \emptyset & (\text{その他}) \end{cases}$$

この φ は iterator として働く。返り値が \emptyset のときには、ビジー状態であるとして `oh.p.foreach` は T_f が変化するまで待機する。`oh.p` では `oh.p.foreach` のオプションで iterator を引数として渡すことができる。また、親ノードにおける計算結果を利用するために、`oh.p.foreach` の `filter` オプションを併用する。

4.4 最小消去多項式導出に関する実験結果

前節で述べた最小消去多項式導出の並列計算法を数式処理システム Risa/Asir に実装し、実験を行った。実験に用いた環境は以下の通りである。

- CPU: Xeon 2.93GHz (4-core) \times 2
- Memory: 48GB
- OS: Debian GNU Linux 2.6

また、160 次整数行列 A を、以下の条件を満たすように、コンパニオン行列のブロック対角行列に対して、行・列基本変形を繰り返して作成した。

- A の特性多項式 $\chi(z)$ と最小多項式 $\pi(z)$ が、

$$\chi(z) = f_1(z)^4 \cdots f_8(z)^4, \quad \pi(z) = f_1(z)^3 \cdots f_8(z)^3$$

となる既約分解をもつ。

- 各既約因子の次数は 5 である。

このとき、160 次整数行列 A を入力データとして、次のタイミングデータを得た。

並列数	1	2	4	6	8
時間 (s)	40.64	31.00	16.93	12.12	9.45

表 1: 160 次整数行列に対する最小消去多項式導出

このことから以下のことが観察される。

- 計算時間は並列数にほぼ比例する。
- 並列数 8 でおよそ 4 倍の性能である。

参 考 文 献

- [1] 小原功任, 田島慎一: 行列のスペクトル分解・固有ベクトルの分散計算, 京都大学数理解析研究所講究録 **1666**(2008), 65–68.
- [2] K. Ohara and S. Tajima: Spectral Decomposition and Eigenvectors of Matrices by Residue Calculus, Proceedings of the Joint Conference of ASCM 2009 and MACIS 2009, COE Lecture Note **22** (2009), Kyushu University, 137–140.
- [3] 加藤涼香, 田島慎一: 有理関数のローラン展開アルゴリズムと代数的局所コホモロジー, 京都大学数理解析研究所講究録 **1395** (2004), 50–56.

- [4] 庄司卓夢, 田島慎一: 1 変数代数的局所コホモロジー類に対する Risa/Asir 用パッケージ `taji_alc`, Risa/Asir Journal **2** (2007), 1–32.
- [5] 田島慎一, 飯塚由貴恵: 行列のスペクトル分解アルゴリズムについて, 京都大学数理解析研究所講究録 **1666** (2008), 49–56.
- [6] 飯塚由貴恵, 田島慎一: 行列のスペクトル分解アルゴリズム—最小多項式が重複因子を持つ場合, 数式処理 **16-2** (2009).
- [7] 飯塚由貴恵, 田島慎一: 行列のスペクトル分解アルゴリズム—最小多項式が複数の重複因子から成る場合, 京都大学数理解析研究所講究録掲載予定.
- [8] 田島慎一, 奈良洸平: 最小消去多項式候補とその応用, 京都大学数理解析研究所講究録投稿中.
- [9] 小原功任: OpenXM を用いた Risa/Asir 並列計算フレームワークの開発, 数式処理 **18-1** (2011).